

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in this application:

1. (Currently amended) A method for updating a seed file to match a target file, said method comprising:

separating said target file into a header portion and a target file payload;

generating target file checking data for one or more blocks of said target file payload, wherein the generating includes, for each of the one or more blocks of said target file payload, retrieving the block from memory, decompressing the block of said target file payload, and calculating file checking data based on the decompressed block of said target file payload;

storing at least a portion of said target file checking data in a cache, wherein the cache is part of a non-volatile storage device;

receiving seed file checking data corresponding to one or more blocks of said seed file, wherein said seed file checking data is based on a decompressed version of the one or more blocks of said seed file;

comparing said seed file checking data with said target file checking data to identify differences in blocks of said seed file and blocks of said target file; and

transmitting information for revising seed file blocks which are different from target file blocks such that said seed file blocks match said target file blocks.

2. (Original) The method of claim 1, wherein said target file checking data and said seed file checking data each comprise weak level checking data and strong level checking data, and wherein said comparing comprises comparing said weak level checking data and next comparing strong level checking data only if a match is identified in said weak level checking data.
3. (Original) The method of claim 1, wherein said target file checking data and said seed file checking data each comprise a 32-bit checksum and a 128-bit checksum.
4. (Original) The method of claim 1, wherein said target file checking data and said seed file checking data each comprise weak level checking data and strong level checking data, and wherein

said storing comprises storing said weak level checking data associated with said target file and storing only said strong level checking data associated with said target file expected to match strong level checking data associated with said seed file.

5. (Original) The method of claim 1, wherein said target file checking data and said seed file checking data each comprise a checksum.
6. (Original) The method of claim 1, wherein said target file checking data stored in a cache are used with multiple updating requests received from a plurality of clients.
7. (Cancel)
8. (Currently amended) The method of claim 1, wherein said seed file ~~and said target file are~~ is decompressed prior to said generating, wherein said seed file blocks are revised in accordance with said transmitted information to match said target file blocks, and wherein said revised seed file blocks are recompressed after revising.
9. (Original) The method of claim 8, wherein said seed file comprises a compressed payload, previously separated from a compound file, and wherein said revised seed file is appended to a header file after said recompressing to constitute a revised compound file.
10. (Original) The method of claim 9, wherein said compound file and said revised compound file comport with an RPM Package Manager format.
11. (Currently amended) A method for updating a seed file to match a target file, said method comprising:
 - constructing said seed file to maximize similarities with said target file;
 - generating seed file checking data for one or more blocks of said seed file, wherein said seed file checking data is based on a decompressed version of the one of the one or more blocks of said seed file;

transmitting said seed file checking data for comparison against cached target file checking data corresponding to one or more blocks of said target file to identify differences in blocks of said seed file and blocks of said target file, wherein the cached target file checking data is cached in a cache that is part of a non-volatile storage device; and

receiving information for revising seed file blocks which are different from target file blocks such that said seed file blocks match said target file blocks.

12. (Currently amended) The method of claim 11, further comprising:

decompressing said seed file prior to said generating seed file checking data;

revising said seed file blocks in accordance with said information to match said target file blocks; and

recompressing said revised seed file blocks.

13. (Original) The method of claim 12, wherein said seed file comprises a compressed payload, previously separated from a compound file, and wherein said revised seed file blocks are appended to a header file after said recompressing to constitute a revised compound file.

14. (Original) The method of claim 13, wherein said compound file and said revised compound file comport with an RPM Package Manager format.

15. (Currently amended) A computer program product, residing on a computer-readable medium, for use in updating a seed file to match a target file, said computer program product comprising instructions for causing a computer to:

separate said target file into a header portion and a target file payload;

decompress said target file payload

generate target file checking data for one or more blocks of said target file payload;

store at least a portion of said target file checking data in a cache, wherein the cache is part of a non-volatile storage device;

receive seed file checking data corresponding to one or more blocks of said seed file;

compare said seed file checking data with said target file checking data to identify differences in blocks of said seed file and blocks of said target file; and

transmit information for revising seed file blocks which are different from target file blocks such that said seed file blocks match said target file blocks.

16. (Original) The computer program product of claim 15, wherein said target file checking data and said seed file checking data each comprise weak level checking data and strong level checking data, and wherein said computer program product further comprises instructions for causing said computer to compare said weak level checking data and to compare said strong level checking data only if a match is identified in said weak level checking data.

17. (Original) The computer program product of claim 15, wherein said target file checking data and said seed file checking data each comprise weak level checking data and strong level checking data, and wherein said computer program product further comprises instructions for causing said computer to store said weak level checking data associated with said target file and to store only said strong level checking data associated with said target file expected to match strong level checking data associated with said seed file.

18. (Original) The computer program product of claim 15, wherein said target file checking data and said seed file checking data each comprise a checksum.

19. (Currently amended) A computer program product, residing on a computer-readable medium, for use in updating a seed file to match a target file, said computer program product comprising instructions for causing a computer to:

construct said seed file to maximize similarities with said target file;

generate seed file checking data for one or more blocks of said seed file;

transmit said seed file checking data for comparison against cached target file checking data corresponding to one or more blocks of said target file to identify differences in blocks of said seed file and blocks of said target file, wherein the cached target file checking data is cached in a cache that is part of a non-volatile storage device; and

receive information for revising seed file blocks which are different from target file blocks such that said seed file blocks match said target file blocks.

20. (Cancel)

21. (New) The method of claim 11, wherein said step of constructing said seed file utilizes data that is locally stored.

22. (New) The method of claim 21, wherein said seed file is a compound file comprising existing versions of individual files.

23. (New) A system for updating a seed file to match a target file, said system comprising:

a client computer that generates a seed file constructed to maximize similarities with said target file, and that generates seed file checking data corresponding to one or more blocks of said seed file;

a server that generates target file checking data for one or more blocks of said target file, receives said seed file checking data, compares said seed file checking data with said target file checking data to identify differences in blocks of said seed file and blocks of said target file and transmits information to said client computer for revising seed file blocks which are different from target file blocks such that said seed file blocks match said target file blocks; and

a non-volatile storage device associated with said server to store at least a portion of said target file checking data in a cache.